

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)特許出願公開番号

特開平5-281999

(43)公開日 平成5年(1993)10月29日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	FI	技術表示箇所
G10L 9/14	G	8948-5H		
	J	8946-5H		
9/18	E	8946-5H		

審査請求 未請求 請求項の数1(全5頁)

(21)出願番号 特願平4-80122

(22)出願日 平成4年(1992)4月2日

(71)出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72)発明者 森尾 智一

大阪府大阪市阿倍野区長池町22番22号

シャープ株式会社内

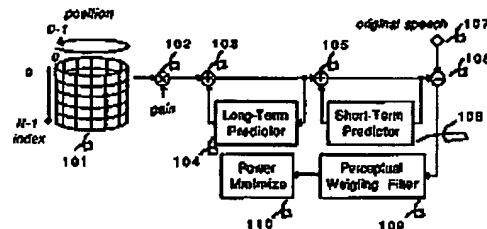
(74)代理人 弁理士 梅田 勝

(54)【発明の名称】 巡回符号帳を用いる音声符号化装置

(57)【要約】

【目的】 符号励振波形予測符号化方式の励振符号帳を記憶するメモリー量と、最適励振符号の探索処理量を削減し、高音質の符号化音声を再生する音声符号化装置を提供する。

【構成】 符号励振波形予測符号化方式を用いる音声符号化装置において、励振波形の符号帳を巡回的にアクセスすることを特徴とし、符号化器における演算量と、符号帳を記憶するメモリーサイズを削減する音声符号化装置。



(2)

特開平5-281999

1

2

【特許請求の範囲】

【請求項1】 符号励振線形予測符号化方式を用いる音声符号化装置において、励振波形の符号帳を巡回的にアクセスすることを特徴とする音声符号化装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、音声波形を効率的に情報圧縮して伝送或いは蓄積する装置に関する。

【0002】

【従来の技術】音声波形を効率的に情報圧縮して、伝送或いは蓄積する方式として、符号励振線形予測符号化(Code-Excited Linear Prediction: 以後CELTPと記す)がある。(例えば、"Code-Excited Linear Prediction (CELTP): High-Quality Speech at Very Low Bit Rates", M. R. Schroeder and B. S. Atal, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 937-940, 1985)。

【0003】CELTPは低ビットレートでも高音質の音声再生できる方式ではあるが、基本的には合成による分析手法(Analysis-by-Synthesis)を用いて符号化するゆえ、演算量が非常に多いという問題がある。特に演算量の多い部分は、最適な励振波形を多種類記憶された符号帳を全探索して決定する処理である。

【0004】この励振符号帳の探索演算量を削減する方法が種々提案されている。符号帳にある構造を規定することで演算量を削減する方式として、ベクトル加算によるコードブック(例えば、"Vector Sum Excited Linear Prediction (VSELP) Speech Coding at 8 kbps", I. A. Gerson and M. Jastuk, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 461-463, 1990)や、オーバーラッピングコードブックがある(例えば、"Improved Speech Quality and Efficient Vector Quantization in SELP", W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 155-158, 1988)。

【0005】本発明は、基本的にはオーバーラッピングコードブックの手法の改良である。以下にW. B. Kleijnらの前述の参考文献に従って、オーバーラッピングコードブックを用いる演算量の削減について説明を進める。

【0006】現分析フレームの音声信号のベクトルをs、合成フィルタのインパルス応答を表す行列をH、現フレームの合成フィルタの零入力応答をzと表すと、最適な励振波形ベクトルtは数1の関係を満たす。

【0007】

【数1】

$$s = Ht + z$$

【0008】ここでHは、その合成フィルタのインパルス応答をRサンプルで打ち切って数2のように表現する。

【0009】

【数2】

$$H = \begin{pmatrix} h_0 & 0 & \dots & 0 \\ h_1 & h_0 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ h_{R-1} & h_{R-2} & \dots & \vdots \\ 0 & h_{R-1} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \vdots & \dots & 0 & h_{R-1} \end{pmatrix}$$

【0010】最適励振波形の探索は、最適ゲイン情報で増幅された候補ベクトル r_i の中から、数3で示される ϵ を最小化するものを全てのi(コードワードの種類だけある)から選択する処理である。

【0011】

【数3】

$$\epsilon = (t - r_i)^T H^T H (t - r_i)$$

【0012】数3は数4のように展開される。

【0013】

【数4】

$$\epsilon = t^T H^T H t - 2r_i^T H^T H t + r_i^T H^T H r_i$$

【0014】数4の右辺第1項はtによらず一定、第2項の $H^T H t$ はtによらず一定で、結局励振波形ベクトルとの内積演算になる。処理量の多いのは第3項目である。この3項目の再帰処理について以下に述べる。

【0015】ここで数5、数6で表されるシフト行列と、マスク行列を定義する。

【0016】

【数5】

$$S = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$

【0017】

【数6】

$$L_k = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & 1 & \vdots \\ \vdots & \vdots & \vdots & \vdots & 0 \end{pmatrix}$$

【0018】 L_k は、k行目に最後の符号要素の現れる行列である。これらの行列を用いるとオーバーラッピングコードブックでは励振波形は数7のように再帰的に表現できる。

【0019】

【数7】

(3)

特開平5-281999

3

4

$$r_{i+1} = S r_i + (1 - I_{N-1}) r_{i+1}$$

* 数8のように再帰的に表現できる。

【0020】処理量の多い数4の第3項目 $E_i = r_i^T H H_i$ 【0021】

は、Hが対称テブリッツ行列であることより、結局、* 【数8】

$$E_{i+1} = E_i + 2[r_{i+1}^T (1 - I_{N-1}) H^T H S r_i - r_i^T (1 - I_1) H^T H_1 r_i] - r_i^T I_1 H^T H_1 r_i + r_{i+1}^T (1 - I_{N-1}) H^T H (1 - I_{N-1}) r_{i+1}$$

【0022】数8の演算は符号の要素が少なく、結局2R+3回の積和演算になる。

※を削減する。

【0023】

【0028】

【発明が解決しようとする課題】VSELPを用いた基底ベクトルを+1或いは-1倍して加算することで、2個の励振波形を作り出す。これにより処理量とメモリー量削減を実現している。例えば512個の励振波形は9個の基底ベクトルから作られる。しかしながら次の問題点がある。

【作用】符号励振波形予測符号化方式を用いる音声符号化装置において、励振波形の符号帳は1個或いは複数個の基底ベクトルを記憶しており、各々の波形を巡回的にシフトしながらアクセスし励振波形を作成する。オーバーラッピングコードブックと同様に再帰的処理により、最適励振波形探索の演算量が削減できる。

【0024】励振波形のベクトルの次元数を40次元とする。この40次元の超空間を512個の代表ベクトルでベクトル量子化する訳だが、9個の基底ベクトルの加減算で作られるベクトルは、高々9次元の空間上に張られるのみであって、40次元の空間を代表するには片寄りが大きいという問題がある。これは符号化音声の品質劣化の原因となり得る。

【0029】

【実施例】

【0025】またベクトル加算によって生成される各励振波形のエネルギーを一定に保つことが困難ゆえ、励振波形のエネルギーを表現する情報量が多く必要となる問題点もある。

<実施例1>図1は、本発明の巡回符号を備えるCELPの音声符号化装置である。101は本発明の巡回励振符号帳。102は励振波形をゲイン (gain) 情報で増幅する掛算器。103及び104は加算器。105は音声のピッチ構造を生成する長期予測器。106は音声のスペクトル構造を生成する短期予測器。107は音声信号の入力端子。108は入力音声信号と合成音声信号の減算器。109は入力音声信号と合成音声信号の差信号の聴覚重み付けフィルタ。110は聴覚的加重付けされた差信号のエネルギー最小化判定器である。

【0026】一方オーバーラッピングコードブックは、コードブックとして長いサンプル長の励振波形をシフトしてアクセスすることで、演算量とメモリー量削減を実現している。例えばサンプル長が511の励振波形を1ずつシフトしてアクセスすることで、40次元のコードブックが512個作成できる。しかしながら、この処理法では生成される各励振波形のエネルギーが一定でないという問題点がある。

【0030】先ず励振波形の生成方法について説明し、その後処理量削減について説明する。励振符号帳101には複数の励振波形が記憶されている。波形のサンプル長 (次元数) をD、個数をNとする。記憶されている各々の波形を $E_i(n)$ 、 $i = 0, 1, \dots, N-1$ 、 $n = 0, 1, \dots, D-1$ と表す。この巡回符号帳より生成される符号は、励振波形を選択するパラメータインデックス (index) と、そのシフトアクセス位置を示すパラメータポジション (position) により、数9により励振波形を生成する。

【0027】

【0031】

【課題を解決するための手段】符号励振波形予測符号化方式を用いる音声符号化装置において、励振波形の符号帳を巡回的にアクセスする手段を備えることで、符号化器における演算量と、符号帳を記憶するメモリーサイズ※

【数9】

$$E_{index, position}(n) = E_{index}((position + n) \% D)$$

【0032】ここで、 $n = 0, 1, \dots, D-1$ 、 $a \% b$ は、aをbで割った余りを示す。

【0034】またエネルギーを正規化した互いに直交するランダムコードブックの直交基底ベクトルに選ぶこともできる。これはホワイトガウシアン (white Gaussian) から生成されるランダムコードブックを、グラムシュミット (Gram-Schmidt) の直交化手法を用いて、全ての他の基底ベクトルに対して直交化した後、エネルギーを正規化することで得られる。

【0033】例えば40次元の励振波形16種類を記憶した励振波形それぞれを、32個の異なるシフト位置から巡回的にアクセスすることで、512個の励振波形が生成できる。励振符号帳に記憶されている複数の励振波形の作成方法は、種々考えられる。例えば単純にはエネルギーを正規化したランダムコードブック (ランダムコードブックについては前述のM. R. Schroeder の文献参照) が用いられる。

【0035】このようなホワイトガウシアン (white Gaussian) の直交基底ランダムコードブックを巡回的にシフトして生成される励振波形ベクトルは、互いに成す角

(4)

特開平5-281999

5

6

度が直交に近く、ランダムコードブックとして表現される励振波形のベクトル空間を、ベクトル量子化する点では効率的であると考えられる。また励振波形ベクトルのエネルギーも全て正規化されるので、励振波形のエネルギーを表現する情報（図1中ではgainと表記）も効率的に表現できる。

【0036】次に巡回符号帳の中から最適な励振波形を探索する際の、処理量削減について述べる。基本的にはオーバーラッピングコードブックで用いられる手法に基づくが、巡回的にアクセスすることで再帰処理の部分が更に処理量削減できる。従来技術で説明したオーバーラッピングコードブックを用いる演算量削減の手順を、巡回符号に適用してみる。ここで数10で表される行列を定義する。

【0037】

【数10】

$$H^T H = A = \begin{pmatrix} a_0 & a_1 & \dots & a_{D-1} \\ a_1 & a_0 & a_1 & \dots & a_{D-2} \\ \dots & \dots & \dots & \dots & \dots \\ a_{D-1} & \dots & \dots & a_1 & a_0 \end{pmatrix}$$

【0038】ここで、 D はベクトル次元数、即ち分析フレームのサンプル数を表す。巡回的に生成される励振波形は数1で表されるが、以下表記を簡潔にするため、 r_i は $E_{a_0, \dots, a_{D-1}}(n)$ 、 $n = 0, 1, \dots, D-1$ を表すものとする。また再帰処理の一側として $i=0$ で以下説明するが、 i は0, 1, ..., $D-1$ のいずれの値でも良い。巡回符号の場合について数9の右辺を $r=0$ とし、実際の演算を記述すると、

【0039】

【数11】

$$r_1^T (I - I_{N-1}) H^T H S r_0 = \sum_{i=1}^{D-1} a_{D-i} r_i$$

【0040】

【数12】

$$r_0^T (I - I_1) H^T H I_1 r_0 = \sum_{i=1}^{D-1} a_i r_i$$

【0041】

【数13】

$$r_0^T I_1 H^T H I_1 r_0 = a_0 r_0^2$$

【0042】

【数14】

$$r_1^T (I - I_{N-1}) H^T H (I - I_{N-1}) r_1 = a_0 r_0^2$$

【0043】結局、巡回符号を用いることで数9の演算は数15で表され、 $D+2$ 回の積和演算になる。

【0044】

【数15】

$$E_1 = E_0 + 2r_0 \left(\sum_{n=1}^{D-1} b_n r_n \right)$$

【0045】ここで、 $b_n = a_{D-n} - a_0$ であり、 H が決定すると一度計算するのみである。再帰処理の演算量は、オーバーラッピングコードブックの場合、インパルスレスポンスの打ち切り数 R に、そして巡回符号の場合は分析フレームのサンプル数 D に依存する。処理量の具体的な比較例として、 $R = D = 40$ とすると、 $(D+2)/(2R+3)$ の比率は約50%になる。

【0046】しかしながら、巡回符号の生成源となる符号を複数種類(N)備える場合は、数9の再帰処理の初期値 E_0 の計算を複数回行わねばならない。それゆえ再帰演算以外の処理量は増加する。

【0047】

【発明の効果】以上より明らかなように、符号励振線形予測符号化方式を用いる音声符号化装置において、エネルギーを正規化した直交基底のランダムコードブックからなる励振波形の符号帳を、巡回的にアクセスすることで、励振波形を効率良くベクトル量子化することができるとともに、励振波形を記憶するメモリー量を削減し、最適励振波形探索の演算量も削減できる。

【図面の簡単な説明】

【図1】本発明の音声符号化装置の符号化器の一実施例のブロック図である。

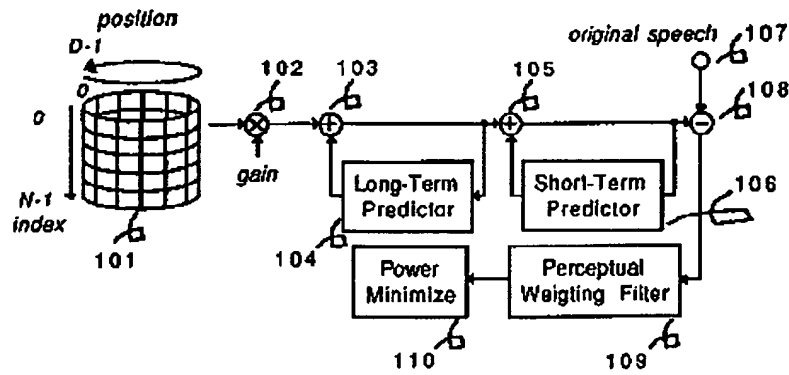
【符号の説明】

- 101 巡回励振符号帳
- 102 掛算器
- 103、105 加算器
- 104 長期予測器
- 106 短期予測器
- 107 信号入力端子
- 108 減算器
- 109 聴覚的重み付けフィルタ
- 110 エネルギー最小化判定器

(5)

特開平5-281999

【図1】



特開平5-281999

【公報種別】特許法第17条の2の規定による補正の掲載
 【部門区分】第6部門第2区分
 【発行日】平成9年(1997)1月10日

【公開番号】特開平5-281999
 【公開日】平成5年(1993)10月29日
 【年通号数】公開特許公報5-2820
 【出願番号】特願平4-80122
 【国際特許分類第6版】

G10L 9/14

9/18

【F1】

G10L 9/14 G 9381-SH
 J 9381-SH
 9/18 E 9381-SH

【手続補正書】

【提出日】平成8年1月12日

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】0038

【補正方法】変更

【補正内容】

【0038】ここで、Dはベクトル次元数、即ち分析フレームのサンプル数を表す。巡回的に生成される励振波形は数1で表されるが、以下表記を簡潔にするため、 r はEindex、 (n) 、 $n=0, 1, \dots, D-1$ を表すものとする。また再帰処理の一例として、

=0で以下説明するが、 i は0、1、 \dots 、 $D-1$ のいずれの値でも良い。巡回符号の場合について数8の右辺を $i=0$ とし、実際の演算を記述すると、

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0043

【補正方法】変更

【補正内容】

【0043】結局、巡回符号を用いることで数8の演算は数15で表され、 $D+2$ 回の積和演算になる。